Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: 31 December 2025

D. Wing
Citrix

29 June 2025

A Referee to Authenticate Servers in Local Domains
draft-wing-settle-referee-01

Abstract

Obtaining and maintaining PKI certificates for devices in a local domain network is difficult for both technical and human factors reasons. This document describes an alternative approach to securely identify and authenticate servers in the local domain using a HTTPS-based trust anchor system, called a Referee. The Referee allows bootstrapping a network of devices by trusting only the Referee trust anchor in the local domain.

About This Document

This note is to be removed before publishing as an RFC.

The latest revision of this draft can be found at https://danwing.github.io/referee/draft-wing-settle-referee.html. Status information for this document may be found at https://datatracker.ietf.org/doc/draft-wing-settle-referee/.

Discussion of this document takes place on the SETTLE mailing list (mailto:settle@ietf.org), which is archived at https://mailarchive.ietf.org/arch/browse/settle/. Subscribe at https://www.ietf.org/mailman/listinfo/settle/.

Source for this draft and an issue tracker can be found at https://github.com/danwing/referee.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Copyright Notice

Table of Contents

1. Introduction

Most existing TLS communications require the server obtaining a certificate signed by a Certification Authority trusted by the client. Within a local domain network this is fraught with complications of both human factors and technical natures (e.g., local domain firewall, lack of domain name).

This document describes a trust anchor system to authorize the legitimate servers on the local domain. The trust anchor host, called a Referee, helps clients identify and authenticate previously-enrolled servers within the local domain. The Referee system purposefully avoids Public Key Infrastructure using X.509 [PKIX], instead using an "allow list" of public keys.

When clients TLS connect to a server on the local domain and encounter a self-signed certificate that might otherwise cause an authentication failure (typically, a warning to the user), the client can send an HTTP query the local domain's pre-authorized Referee system to learn if that server has been enrolled with the Referee. If so, it indicates the server was enrolled in the Referee trust anchor and the TLS connection can continue.

2. Unique Characteristics

The system described in this draft has several characteristics that differ from other trust anchor systems:

* requires an always-on Referee server to authenticate servers on the local domain,

* the client validates a server is authorized on the local domain via an HTTPS query to the (Referee) server on the local domain, rather than a signed certificate,

* can use raw public keys, as the dates and certificate signatures of servers on the local domain are ignored by this system, in favor of consulting the Referee,

* handles name collisions for servers on different networks, so two different networks can both have servers with the same name (e.g., router.local),

* handles unique names for servers (e.g., router-abcdef123456.local),

* servers that participate in the Referee system can change their public keys periodically and inform the Referee, which allows clients to automatically handle those public key changes, and

* can operate without changes to non-Referee servers on the local domain, provided such servers do not change their public keys.

3. Requirements Evaluation

Using requirements from [I-D.rbw-home-servers], the proposal in this document has the following summarized characteristics:

Solution Name	Reduce CA	Eliminate CA	Existing CA Support	Existing Client Support	Revoke Auth
Referee	Yes	Yes	N/A	No	Yes

Table 1: Summary of Referee Against Requirements

4. Operation

Figure 1 shows a client receiving the DHCP message of the local network’s Referee, connecting to that Referee and, because this Referee has never been seen before by this client, prompting the user if this Referee is to be trusted.

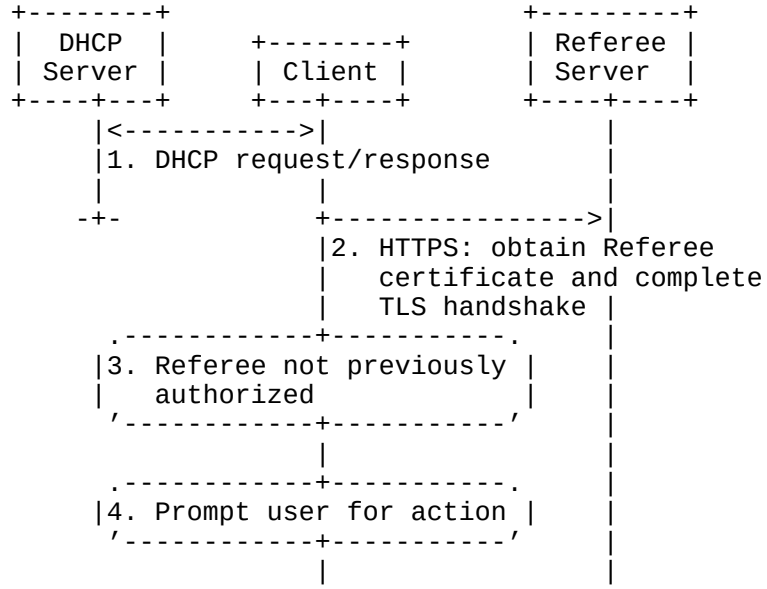

Figure 1: Message Sequence Diagram with New Referee

Figure 2 shows a client connecting to a Referee that was previously authorized by the client. In this case, the user is not prompted to re-authorize the Referee.

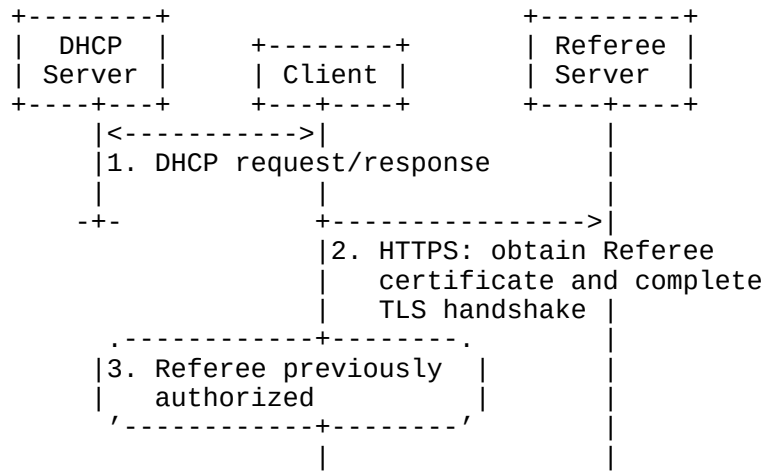

Figure 2: Message Sequence Diagram with Previously-Authorized Referee

Figure 3 shows a client, after performing the above steps with its Referee, connecting to a server on the local domain and then validating that server's public key with its Referee. The validation with the Referee is done in lieu of validating the certificate of that server on the local domain.

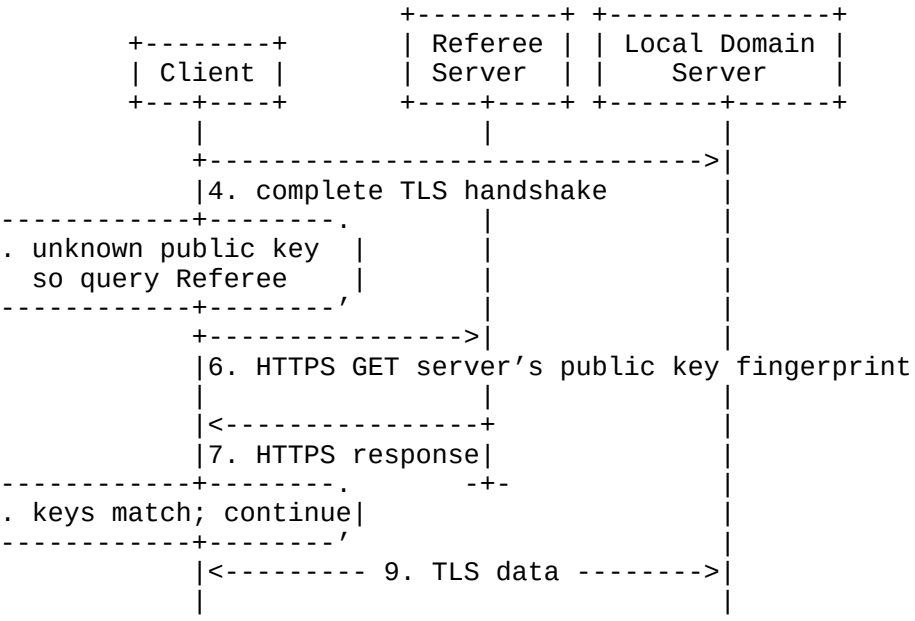

Figure 3: Message Sequence Diagram to Server on Local Domain

4.1. Referee

The Referee trust anchor function is implemented within any always-on device within the local domain (e.g., router, smart home hub, NAS, or a virtualized CPE). The Referee runs HTTPS and serves files containing public key fingerprints indexed by each server's local domain name. These files are populated by servers on the local domain that support Referee or manually as described in Section 5.2.

Clients authenticate the Referee and use HTTP GET to fetch the named public key fingerprint from the Referee server using a well-known URI. The Referee returns the SHA-256 fingerprint of the server's public key as an octet-stream.

For example if the server's name is "smarttv-abcdef123.internal" the following HTTP GET would be issued by the client to retrieve that server's public key fingerprint:

```
GET /.well-known/referee/sha256/smarttv-abcdef123.internal
```

4.2. Servers

A server supporting this specification is expected to be a printer (using IPPS or HTTPS), file server (e.g., NAS or laptop), IoT device, router (especially its HTTPS-based management console), scanner, Smart TV, or similar.

Each local domain device supporting Referee has a public key. During installation of the device to a Referee network, the device's hostname and public key fingerprint are stored into the Referee Server. Several options exist for this step, detailed in Section 5.

If a server's public key changes (e.g., factory reset, key rotation, public key algorithm change) the new key needs to be enrolled with the Referee and the old key removed (see Appendix A.5). Clients will notice the mismatch and will query the Referee, which provides the new key's fingerprint, authenticating the server (with its new key) to the client.

4.3. Clients

A client supporting this specification is first configured with the DNS name of its Referee server, which might occur via service discovery (see Section 7). The client authenticates and authorizes the Referee server using one of the bootstrapping mechanisms (see Section 5). This step occurs only once for each home network the client joins, as each home network is responsible for being a Referee for its own local domain. A quality implementation may reduce user prompting by sharing known Referee server identities across a user's various devices, or by other means.

On a connection to a server on the local domain (see Section 6) the client includes the server's local domain name in the TLS Server Name Indication (SNI) extension of its ClientHello. A client may additionally cache the association of authorized servers to that same local domain, after the client has completed a TLS handshake to the Referee to verify the client is connected to that Referee's network. Upon disconnection from that network, the client invalidates its cache until connected to a new network and validating that network's Referee.

On receiving the server's certificate in the TLS exchange, the client will have previously cached that server+Referee combination, or not, as discussed below:

* If not previously cached, the client queries that network's Referee with the DNS name of the server (e.g., printer.internal). The Referee responds with the public key fingerprint of that server. The client checks if the public key fingerprint (from the Referee) matches the public key of the server (from the TLS handshake). If they match, communication with the server continues and the server name and its public key might be cached by the client. If they do not match, the client aborts this communication session; further actions by the client are an implementation detail.

* If previously cached, the client determines if the cached public key matches the public key obtained from the TLS handshake with the server. If they match, communication continues. If they do not match, the queries the Referee to learn if a new key fingerprint is available for that server. If a different fingerprint is returned by the Referee, the client verifies it matches the public key from the TLS handshake. If they match, the client replaces the information in its cache.

Internally, a client might form a unique identity for a local domain server as hostname (e.g., printer.local) combined with the identity of the Referee, such as the Referee's public key fingerprint (if not signed by a global Certification Authority) or the Referee's name (if signed by a global Certification Authority). In this way, when the client is on a different network (which will have a different Referee), a server name collision (e.g., router.local) will result in a unique internal identity for that server -- keeping all the server-specific data separate for those two servers on different networks (e.g., web forms, passwords, local storage, etc.)

4.4. Revoking Authorization

When the administrator revokes authorization for a server (e.g., replacement of a server), the administrator removes the old public key from the Referee and installs the new key in the Referee.

When this replacement occurs, the clients that have not already cached the server's public key will simply query the Referee, which has the server's new public key. The clients that have cached the server's previous public key will notice the mismatch, pause their communication with the server, and validate with the Referee that the new key is legitimate, and continue their communication with the server.

Thus, revoking authentication has immediate effect because the clients immediately validate a mismatch with the Referee.

5. Bootstrapping the Referee

5.1. Clients to Referee

The clients have to be configured to trust their Referee. This is a one time activity, for each home network the client joins. This can be somewhat automated using service discovery (Section 7).

The client is configured to trust the Referee server's public key. If this key changes, session resumption might be useful to avoid having the client re-configured to trust that new public key, see Appendix A.5.2.

5.2. Servers to Referee

Server names and their associated public key fingerprints have to be enrolled with the Referee. This can be automated by servers that support Referee, or can be done manually for servers that do not (yet) support Referee -- providing immediate value to Referee clients without waiting for server Referee support.

5.2.1. Short Code or Scan Code

Short code printed on the Referee-capable server which can be scanned by a smartphone application by the home administrator which is authorized to push new associations to the Referee.

5.2.2. Incremental Deployment and Manual Referee Configuration

It is useful for a Referee server to provide immediate value on its installation, even when servers do not (yet) support Referee. The Referee system requires support of both the client (to ask the Referee for mediation) and installation of a Referee -- which could be in the home router, NAS, or other always-on device. This section explores how to bootstrap Referee system when servers on the local domain do not (yet) support Referee.

The Referee has a user interface for manual addition of a server. For example the user might cause the Referee to connect to a server on the local domain using TLS, extract its public key, and create the hostname and public key fingerprint association on the Referee. Additionally, the Referee might also scan the local domain network looking for TLS servers on common ports (e.g., HTTPS, IMAPS, IPPS, NNTPS, IMAPS, POP3S) to enumerate a list of servers for the user to approve the same association on the Referee.

To accommodate servers that change their public key but do not (yet) register that change with the Referee, the Referee can refresh its server fingerprints at user request. The user request might be initiated by the administrator or an HTTP message from the client to the Referee of a key mismatch.

6. Identifying Servers as Local

This section defines the domain names and IP addresses considered "local".

6.1. Local Domain Names

The following domain name suffixes are considered "local":

* ".local" (from [mDNS])
* ".home-arpa" (from [Homenet])
* ".internal" (from [I-D.davies-internal-tld])
* both ".localhost" and "localhost" (Section 6.3 of [RFC6761])

6.2. Local IP Addresses

Additionally, if any host resolves to a local IP address and connection is made to that address, those are also considered "local":

* 10/8, 172.16/12, and 192.168/16 (from [RFC1918])
* 169.254/16 and fe80::/10 (from [RFC3927] and [RFC4291])
* fc00::/7 (from [RFC4193])
* 127/8 and ::1/128 (from Section 3.2.1.3 of [RFC1122] and [RFC4291])

7. Service Discovery

To ease initial bootstrapping the client, the local domain can advertise its Referee server using a new DHCP option (see Section 10). The client connects to that server using HTTPS and extracts the public key. Each local domain has its own Referee which only has purview over servers in its same local domain. The Referee’s public key has either not been seen before or has been seen before:

* If the public key has not been seen before, the user needs to approve use of that Referee trust anchor for this local domain; the exact method is out of scope of this document.

* If the public key has been seen before, and was previously approved (or previously rejected) by the user, that same user decision is applied again.

8. Operational Considerations

The Referee has to always be available. The client cache helps reduce load on the Referee but new clients (e.g., new devices, guest users, restored devices) and client cache invalidation will always cause some traffic to the Referee.

When the Referee is unavailable, clients behavior devolves to what we have today: servers will need to obtain a real PKI certificate signed by a Certification Authority already trusted by the clients, or else clients will need to manually trust individual certificates.

9. Security Considerations

TODO: expand security considerations.

See Section 8 describing client behavior when the Referee is unavailable.

10. IANA Considerations

Register new .well_known URI for Referee server.

Register new DHCP option for Referee server.

11. Informative References

[Homenet] Pfister, P. and T. Lemon, "Special-Use Domain 'home.arpa.'", RFC 8375, DOI 10.17487/RFC8375, May 2018, <https://www.rfc-editor.org/rfc/rfc8375>.

[I-D.davies-internal-tld] Davies, K., McConachie, A., and W. Kumari, "A Top-level Domain for Private Use", Work in Progress, Internet-Draft, draft-davies-internal-tld-03, 3 March 2025, <https://datatracker.ietf.org/doc/html/draft-davies-internal-tld-03>.

[I-D.rbw-home-servers]
Reddy.K, T., Boucadair, M., and D. Wing, "Identifying and Authenticating Home Servers: Requirements and Solution Analysis", Work in Progress, Internet-Draft, draft-rbw-home-servers-00, 19 September 2024, <https://datatracker.ietf.org/doc/html/draft-rbw-home-servers-00>.

[mDNS] Cheshire, S. and M. Krochmal, "Multicast DNS", RFC 6762, DOI 10.17487/RFC6762, February 2013, <https://www.rfc-editor.org/rfc/rfc6762>.

[PKIX] Cooper, D., Santesson, S., Farrell, S., Boeyen, S., Housley, R., and W. Polk, "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", RFC 5280, DOI 10.17487/RFC5280, May 2008, <https://www.rfc-editor.org/rfc/rfc5280>.

[RFC1122] Braden, R., Ed., "Requirements for Internet Hosts - Communication Layers", STD 3, RFC 1122, DOI 10.17487/RFC1122, October 1989, <https://www.rfc-editor.org/rfc/rfc1122>.

[RFC1918] Rekhter, Y., Moskowitz, B., Karrenberg, D., de Groot, G. J., and E. Lear, "Address Allocation for Private Internets", BCP 5, RFC 1918, DOI 10.17487/RFC1918, February 1996, <https://www.rfc-editor.org/rfc/rfc1918>.

[RFC3927] Cheshire, S., Aboba, B., and E. Guttman, "Dynamic Configuration of IPv4 Link-Local Addresses", RFC 3927, DOI 10.17487/RFC3927, May 2005, <https://www.rfc-editor.org/rfc/rfc3927>.

[RFC4193] Hinden, R. and B. Haberman, "Unique Local IPv6 Unicast Addresses", RFC 4193, DOI 10.17487/RFC4193, October 2005, <https://www.rfc-editor.org/rfc/rfc4193>.

[RFC4291] Hinden, R. and S. Deering, "IP Version 6 Addressing Architecture", RFC 4291, DOI 10.17487/RFC4291, February 2006, <https://www.rfc-editor.org/rfc/rfc4291>.

[RFC6761] Cheshire, S. and M. Krochmal, "Special-Use Domain Names", RFC 6761, DOI 10.17487/RFC6761, February 2013, <https://www.rfc-editor.org/rfc/rfc6761>.

[RFC8446] Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", RFC 8446, DOI 10.17487/RFC8446, August 2018, <https://www.rfc-editor.org/rfc/rfc8446>.

Appendix A. Issues for Further Discussion

A.1. PKI Fallback

Currently the text suggests clients should fallback to PKI if Referee validation fails. This means certificate warnings for self-signed certificates. Is such fallback harmful or is it worthwhile?

A.2. Distinct Local Domain with its Own Referee

Each local domain is anticipated to have its own Referee. Thus, when a client visits another network, that network will have its own Referee which is learned via service discovery. That Referee is bootstrapped same as the 'home' network's Referee (see Section 5).

A.3. Redundant Referees on One Local Domain

This draft only discusses a single Referee on each Local Domain. Multiple Referees may well be desirable for redundancy but are out of scope of this draft.

A.4. Unique Names

Printer.internal or printer.local are handy names and can be used with a Referee system.

Unfortunately existing browsers have state that is tied to names -- web forms, cookies, and passwords. Thus, those existing systems need names that contain a unique identifier like a UUID, e.g., printer.2180be87-3e00-4c7f-a366-5b57fce4cbf7.internal. Or perhaps embedding part/all of the public key into the name itself, for example:

```
printer.2180be87-3e00-4c7f-a366-5b57fce4cbf7.internal
nas.103a40ee-c76f-46da-84a1-054b8f18ae33.internal
router.fb5f73ed-275a-431e-aecf-436f0c54d69d.local
```

The Referee system is ambivalent about the host name -- the Referee's name and each server's name need only be unique on the current local domain. Name collisions that occur between local domains are handled by the client querying the other network's trusted Referee to check legitimacy.

The Referee system allows keeping the unique name the same for the lifetime of the device while allowing changing its public key, as discussed in the following section.

A.5. Key Lifetime (Rotating Public Key)

For security hygiene, the public keys in a server and the Referee may be occasionally changed. This section discusses how such changes are handled by a Referee system.

A.5.1. Server

If a server's public key changes the new key has to be installed into the network's Referee. To automate such changes, the server could connect to the Referee and prove possession of its (old) private key (using TLS client authentication or using application-layer mechanism such as JSON Web Signature) and publish its new public key using an HTTP PUT.

> Note: such a PUT mechanism also means an attacker in possession of the server's private key can change the legitimate server's public key fingerprint in the Referee to now point at an attacker-controlled system, denying access to the legitimate server. It is still better than unencrypted connections, which is the case today.

A.5.2. Referee

If the Referee's public key changes all the clients have to re-authenticate the Referee's new public key. This is uncool.

To allow changing the Referee's public key without needing the client to re-authenticate the Referee, the client and Referee could do session resumption for its subsequent connections to the Referee (Section 2.2 of [RFC8446]). If session resumption succeeds, the client can query a the Referee's own well-known URI to determine if the Referee's public key has changed, and update itself accordingly.

With the above technique, the client will only have to (manually) re-authenticate the Referee when the Referee cannot perform session resumption. As session resumption is usually implemented using the server's private key, the Referee would need to remember its previous private key (or two or three).

Acknowledgments

Thanks to Sridharan Rajagopalan for reviews and feedback.

Author's Address

Dan Wing
Citrix
United States of America
Email: danwing@gmail.com